

MpSoC, IP und Plattformen oder wie nutzt man 1 Milliarde Transistoren?

Ernst, Rolf

Veröffentlicht in:
Jahrbuch 2002 der Braunschweigischen
Wissenschaftlichen Gesellschaft, S.93-96



J. Cramer Verlag, Braunschweig

ROLF ERNST, Braunschweig

MpSoC, IP und Plattformen oder wie nutzt man 1 Milliarde Transistoren?

Braunschweig, 11.10.2002*

Die exponentielle Zunahme der Komplexität integrierter Schaltungen um den Faktor 3 bis 4 alle 3 Jahre hält seit den 60er Jahren ungebrochen an. So wird etwa für Speicher eine Steigerung der Komplexität von circa 540 Millionen Transistoren pro integrierte Schaltung im Jahr 2001 auf 64 Milliarden Transistoren in 2016 erwartet. Längst hat sich eine ganze Industrie planerisch auf die Kontinuität dieser rasanten Entwicklung eingestellt. In der „ITRS Roadmap“ werden für mehr als ein Jahrzehnt die jährlich zu erzielenden Strukturgrößen und Fertigungsprozesse festgelegt [ITR01].

Weit weniger strukturiert entwickelt sich die Nutzung der daraus verfügbaren Schaltungskapazität. In größeren zeitlichen Abständen bilden sich als „Design Gap“ bezeichnete Umbruchsphasen im Entwurf heraus, in denen die Entwurfsproduktivität der Zunahme der Schaltungskomplexität nicht mehr folgen kann und eine Abkehr von eingeführten Entwurfsmethoden erzwingt. Mittlerweile wurde der Übergang zu einem komponentenbasierten Entwurstil vollzogen, bei dem große Teile früherer Entwürfe nahezu unverändert übernommen und über komplexe Kommunikationsstrukturen miteinander verbunden werden. Darauf aufbauend hat sich eine Zulieferindustrie entwickelt, die Halbleiterfirmen mit Komponentenentwürfen, sogenannten IP-Komponenten, beliefert, welche quasi als Blaupausen übernommen und an die jeweilige Technologie angepasst werden. Bei dieser Arbeitsteilung lohnt es sich, Komponenten zu optimieren, beispielsweise auf einen niedrigen Stromverbrauch, der bei zunehmender Mobilität der Informationstechnik entscheidend für die Wettbewerbsfähigkeit einer integrierten Schaltung wird.

Als Folge dieser Entwicklung weisen neuere Chipentwürfe eine heterogene Struktur aus unterschiedlichen Prozessoren, Coprozessoren und Speichern auf, die zu hochintegrierten „Multiprocessor Systems-on-Chip“ (MpSoC) zusammengesetzt werden. Derartige MpSoC finden sich beispielsweise in neueren Mobiltelefonen, Set-Top Boxen oder in integrierten Netzwerkprozessoren, die ein neues Marktsegment der Telekommunikation bilden.

Entsprechend der Hardware entwickelt sich auch die Softwarearchitektur zu einem heterogenen und vielschichtigen Gefüge von konfigurierbaren Komponenten mit den gegenläufigen Zielen einer effizienten Nutzung der Hardware und einer einfachen Programmierung der Anwendung, also etwa der Sprachverarbeitung in einem Mobiltelefon. Der Korrektheitsnachweis derart komplizierter Hardware-Softwareentwürfe ist zu einem zentralen Problem des Systementwurfs geworden.

* Kurzfassung eines Vortrags gehalten in der Klasse für Ingenieurwissenschaften der Braunschweigischen Wissenschaftlichen Gesellschaft.

Die MpSoC, die in Mobiltelefonen, Fahrzeugen oder in der Konsumelektronik zum Einsatz kommen, zählen zu den eingebetteten Systemen. Darunter versteht man allgemein ein (Mikro-)Computersystem, das in ein technisches System „eingebettet“ ist, das selbst nicht als Computer erscheint. Die Funktionen dieser eingebetteten Systeme definieren heute zu einem großen Teil die Eigenschaften von Industrieprodukten, die ohne sie kaum noch konkurrenzfähig realisierbar wären. Damit lastet eine erhebliche Verantwortung auf dem Entwurf dieser Systeme, die oft sicherheits- oder zeitkritische Funktionen übernehmen, etwa in der Fahrzeugtechnik, der Medizinelektronik oder der Telekommunikation. Der Entwurf unterliegt damit schärferen Qualitätsanforderungen als im Fall von PC-Prozessoren und ihrer Software.

Dem Korrektheitsnachweis solcher Systeme kommt folglich eine zentrale Rolle zu. Dabei wird der implementierungsunabhängige Nachweis der korrekten Funktion eines Systems unterschieden vom Nachweis einer ausreichenden Performanz, d.h. der hardware-abhängigen Einhaltung von Zeitbedingungen, der Vermeidung von Überlastsituationen und von Überläufen der internen Informationsspeicher. In beiden Fällen erforscht man neben dem in der Praxis heute üblichen Korrektheitsnachweis durch Simulation auch formale Beweisverfahren. Während sich die formale Verifikation von Hardwarefunktionen in der Folge des enorm kostspieligen, sogenannten „Pentium bug“ (einem fehlerhaften Rechenwerk im Pentium-Prozessor der Firma Intel) auch schon in der Praxis zunehmend Verwendung findet, steht der sehr komplizierte formale Nachweis der Performanz noch am Anfang seiner Entwicklung. Der Performanznachweis wird erschwert durch die sehr komplizierten Prozessorarchitekturen, die Berechnungsschritte automatisch parallelisieren und damit ihre Ausführungsreihenfolge vertauschen können, und die hierarchischen Informationsspeicher mit enorm komplexer Zugriffssteuerung [Fer+01]. In der Folge können zeitliche Anomalien entstehen, die in bestimmten Situationen plötzlich zu einer erheblichen Verlangsamung der Programmausführung führen können, die durch Simulation kaum erkennbar ist, die aber auch die Entwicklung formaler Modelle vor große Herausforderungen stellt [LuSt99]. Ein zweites Problem ist die Heterogenität der Hardware- und Softwarestrukturen, die zu sehr unterschiedlichen Datenflüssen in einer integrierten Schaltung führt, die wiederum über ein gemeinsames Kommunikationsmedium (einen oder mehrere sogenannte Busse) übertragen werden. Auch hier treten Anomalien auf, die zu sporadischen Ausfällen führen können [RiEr02]. Dies ist gerade bei der Integration von Komponenten ein kniffliges Problem. So wird eine Komponente üblicherweise für den ungünstigsten Fall spezifiziert, in dem die größte Laufzeit auftritt. Sendet diese Komponente aber wiederholt Daten über ein gemeinsames Kommunikationsmedium, so wird bei gleicher Datenmenge pro Ausführung die Last bei kurzer Ausführungszeit am höchsten sein. Der günstigste Fall der Komponenten wird damit zum ungünstigsten Fall des Systems. Noch komplizierter wird der Ablauf bei Softwareprozessen, die sich in ihrer Ausführung unterbrechen, wie dies in der Echtzeitdatenverarbeitung üblich ist, um eine Priorität von Aufgaben mit unterschiedlichen „Deadlines“ zu realisieren.

Ansätze zur formalen Analyse derartiger komplexer MpSoC-Architekturen sind derzeit noch rar. Es gibt „holistische“ Ansätze, die eine geschlossene Darstellung suchen [PEP02], und Ansätze, die mit formaler Kopplung heterogener Komponentenmodelle arbeiten

[CEK+02]. Zur letzteren Gruppe zählen die Arbeiten am Institut für Datentechnik und Kommunikationsnetze [RiEr02, RZJ+02]. Die Grundidee ist dabei eine Kopplung bekannter Analyseverfahren [But02] für individuelle Komponenten zu einer Gesamtanalyse, wobei die Analyseverfahren in der gleichen modularen Weise gekoppelt werden wie die Komponenten selbst. Mit diesem Verfahren können auch sehr komplexe heterogene MpSoC analysiert werden. In verschiedenen Projekten mit der Industrie werden gegenwärtig Anwendungsszenarien dieser Verfahren praktisch erprobt [JRE03].

Die nächste Herausforderung des Entwurfs integrierter Schaltungen liegt im zunehmenden Einfluss der Laufzeiten über die chipinternen Verbindungen auf die Eigenschaften eines MpSoC. Bereits heute liegen die Laufzeiten längerer Leitungen um ein Vielfaches höher als die Schaltzeiten der Transistoren. Dieser Abstand wird künftig weiter wachsen. Man benötigt daher neue Entwurfsverfahren, die Hardwarekomponenten auf ein optimiertes Kommunikationsnetzwerk anpassen und nicht umgekehrt vorhandene Hardwarekomponenten über ein Kommunikationsnetzwerk integrieren, wie dies bislang in der Chipentwicklung der Fall war. Dieser kommunikationszentrierte Entwurf stellt einen neuen Trend mit intensiven Forschungsaktivitäten dar [ITR01, BeDM02]. Dabei greift man gern auf Konzepte der Telekommunikation zurück, der diese Fragestellung lange vertraut ist.

Literatur:

- [BeDM02] L. BENINI & G. DE MICHELI, *Networks on chips: a new SoC paradigm*. IEEE Computer, Vol. **35** (1), pp. 70-78, Jan. 2002.
- [But02] G. BUTTAZZO, *Real-Time Computing Systems - Predictable Scheduling Algorithms and Applications*, Kluwer Academic Publishers, 2002.
- [CEK+02] S. CHAKRABORTY, T. ERLEBACH, S. KÜNZLI & L. THIELE, *Schedulability of Event-Driven Code Blocks in Real-Time Embedded Systems*. In: Proc. 39th Design Automation Conference (DAC02), pp. 161-621, New Orleans, USA, 2002.
- [Fer+01] CH. FERDINAND et al., *Reliable and Precise WCET Determination for a Real-Life Processor*. Embedded Software Workshop, Springer LNCS 2211, pp. 469ff., Lake Tahoe, USA, Oct. 2001.
- [ITR01] J. ADAM et al., *International technology roadmap for semiconductors (ITRS) 2001*, <http://public.itrs.net>.
- [JRE03] M. JERSAK, K. RICHTER & R. ERNST, *Formal Methods for Integration of Automotive Software*. To Appear: Proc. Design Automation and Test in Europe Conference 2003 (DATE03), Munich, March 2003.
- [LuSt99] TH. LUNDQVIST & P. STENSTRÖM, *Timing Anomalies in Dynamically Scheduled Microprocessors*. Proc. 20th IEEE Real-Time Systems Symposium, pp.12ff, Dec. 1999.
- [PEP02] T. POP, P. ELES & Z. PENG, *Holistic Scheduling and Analysis of Mixed Time/Event-Triggered Distributed Embedded Systems*. In: Proc. 10th Internatio-

nal Symposium on Hardware/Software Codesign (CODES02), pp. 187-192, Estes Park, CO, USA, 2002.

[RiEr02] K. RICHTER & R. ERNST, *Event Model Interfaces for Heterogeneous System Analysis*. Proc. Design Automation and Test in Europe Conference 2002 (DATE02), pp. 506-513, Paris, March 2002.

[RZJ+02] K. RICHTER, D. ZIEGENBEIN, M. JERSAK & R. ERNST, *Model Composition for Scheduling Analysis in Platform Design*. Proc. 39th Design Automation Conference (DAC 2002), pp. 854 –861, New Orleans, June 2002.

Prof. Dr.-Ing Rolf Ernst
Ellernbruch 12 B
D-38112 Braunschweig